Users data table

chefs name VARCHAR(100)
Email VARCHAR(50)
Password VARCHAR(500)
PhoneNumber NUMERIC
Experience (years)
Awards VARCHAR(500)
user_id **(Primary Key**)

Recipe data table

Reference to user_id
recipe id SERIAL PRIMARY KEY
recipe_nameVARCHAR(100)
instructions text
public_or_private (boolean)
Reference to ingredient_id

Ingredients data table

Ingredients_id SERIAL PRIMARY KEY
Protein VARCHAR()
Carbohydrates VARCHAR()
Spices VARCHAR()
Veggies VARCHAR()

Occasions data table

Menu Reference to recipe_name
Menu Reference to chef_name
Occasion VARCHAR(500)

- Title this next section “Table Ideas”.
- Based off the ideas you just brainstormed, list out what tables you think you’ll need. It’s okay if you change it up later.
- Write a brief description of each table. For example: “products: this table will hold information about the products offered, each row will be an individual product”.

Table Ideas

I will need a user’s data table, receipt data table, a data table for ingredients in a recipe and an occasions table.

The user table will contain information about the user like there email, years of experience cooking and awards they have received. The recipe data table contains

information about each recipe while the occasions data table contains information about the special occasion for making the recipe.

Step 3

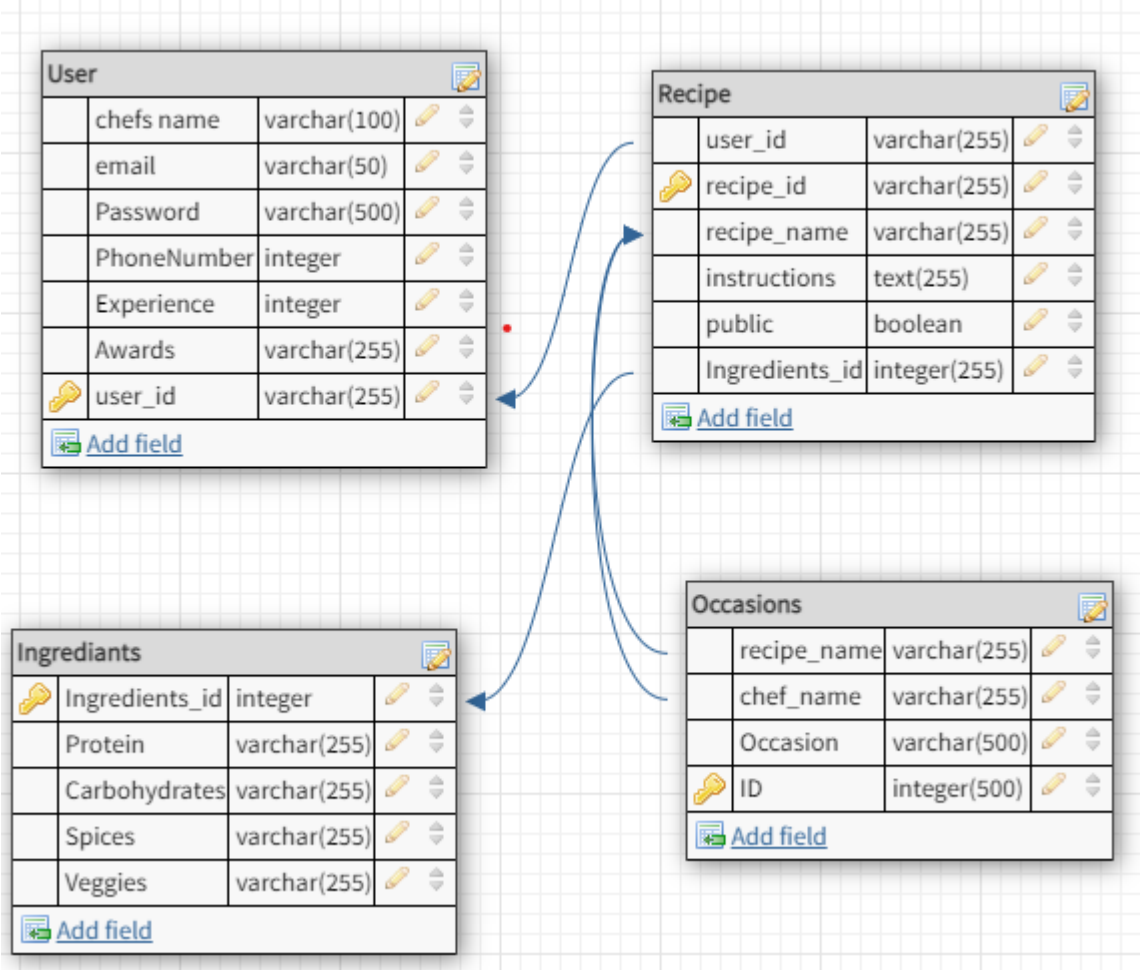
- Next figure out what relationships should exist among the data.
- Title this section “Relationships” and create 3 sub-sections as well – “one-to-one”, “one-to-many”, and “many-to-many”.
- In each subsection, list the tables that have that relationship **and explain why you chose that relationship.**
- For example, let’s say I’m planning an ecommerce app. In the one-to-many section, I could have: “products table and review table because each product can have multiple reviews, but a review is only for one specific product”.
- Note: remember tables can relate to multiple other tables. In the product/reviews example above, reviews would also have a relationship with a users table. And product could be related to a cart table. There could be others as well.

Relationships

In the recipe data table, the user id will be referenced in which comes from the user data table and the ingredient ID will be referenced in the ingredients table. I chose these relationships because it will be important to display this information when each recipe comes up. In the occasions data table the reference to the chef name and recipe name which are included in the user data table and the recipe data table.

Step 4

- Now that you know what relationships you’ll have, go back to your “Table Ideas” and **in a different text color** add in any additional tables and their descriptions that you’ll need.



For the user table I'll be storing all of this information because it is relevant for knowing more about the user and what level they are at when it comes to cooking. For the recipe data table I'll be storing information about each recipe some important ones including instructions. For ingredients ill be storing all the information needed to make a recipe. For the occasions this will target what specific events the users will make the recipes.

```sql
CREATE TABLE user (

  user_id SERIAL PRIMARY KEY,
  chefs_name VARCHAR(100),
  email VARCHAR(50),
  Password VARCHAR(100),
  PhoneNumber NUMERIC,
  Experience NUMERIC,
  AWARDS VARCHAR(200)
);
```

```sql
CREATE TABLE Recipe(

  user_id SERIAL PRIMARY KEY,
  recipe_id VARCHAR(255),
  recipe_name VARCHAR(255),
  instructions TEXT,
  Public BOOLEAN,
  Ingrediants_id NUMERIC

);
```

```sql
CREATE TABLE Ingrediants(

  Ingrediants_id SERIAL PRIMARY KEY,
  Protein VARCHAR(255),
  Carbs VARCHAR(255),
  Spices VARCHAR(255),
  Veggies VARCHAR(255)


);
```

```sql
CREATE TABLE Ocassions(

  ID SERIAL PRIMARY KEY,
  recipe_name VARCHAR(255) NOT NULL REFERENCES recipe(recipe_name),
  Occassion VARCHAR(255)


);
```